



SNACKS ON HIGHWAY

고속도로 휴게소 메뉴 조회 서비스

김혜민



A PRESENTATION ABOUT

- 1 프로젝트 개요
- 2 주제선정 이유
- 3 개발자 소개
- 4 개발일정과 개발과정
- 5 프로젝트 개발 과정
- 6 프로젝트 개발 결과
- 7 자체평가

프로젝트 개요

프로젝트명

-Snacks On Highway

프로젝트 개발자

-김혜민

프로젝트 개발일자

-2022.12.01 ~ 2202 12.14

프로젝트 개발환경

-Visual Studio

-MSSQL



주제선정 이유

고속도로 여행에서 휴게소를 들리는 것은 단순히 급한 용무를 해결하거나 운전 중의 허기를 채우기 위한 간식거리를 위한 것이 대부분이었습니다. 하지만 근간에 들어 각 휴게소만의 특별한 메뉴가 생겨나고 그것을 즐기기 위해서 일부러 휴게소에 들리는 등의 소위 '휴게소 투어'라는 것이 하나의 트렌드이자 소소한 재미로 자리잡고 있는 중입니다.

하지만 이에 대한 정보들은 아직 개인이 검색엔진에서 일일이 검색하여 블로그 등에서 일부의 정보만 접한다는 불편함이 있습니다. 이에 일목요연하게 조회할 수 있고 시즌메뉴나 재료 등등까지도 검색 가능한 휴게소 메뉴 통합검색 프로그램이 있다면 고속도로를 이용한 여행에 편의를 제공할 수 있을 듯 하여 이번 개발주제로 선정하였습니다.

개발자 소개



김혜민

초보 개발자
C# 코드 작성
공공데이터 검색
MSSQL DB 연결
프로그램 구현
발표자료 작성
PPT 발표

개발일정과 개발과정



STEP 1

주제선정
개발방향논의
요구사항파악



STEP 2

공공데이터 수집
카카오API 연결



STEP 3

C# 코드 작성
프로그램 구현
MSSQL DB 연결



STEP 4

PPT 작성
발표 준비

프로젝트 개발과정



- ▶ C# Info.cs
- ▶ introForm.cs
- ▶ C# kakaoAPI.cs
- ▶ C# Locale.cs
- ▶ mainForm.cs
- ▶ marketDataSet.xsd
- ▶ packages.config
- ▶ C# Program.cs

```
namespace ShacksOnHighway
{
    참조 5개
    public partial class MainForm : Form
    {
        //여러 메소드에서 쓰는 변수 선언
        private SqlConnection conn = null;
        private DataTable dt = null;
        DataTable table = new DataTable();
        private List<Info> infoList = new List<Info>();
        private List<Info> infoList2 = new List<Info>();
        private DataTable dgvGridView1 = null;

        참조 1개
        public MainForm() //생성자
        {
            InitializeComponent();

            connectDB();
            inputListBox1();
            inputListBox2();
            //inputListBox3();
            timer1.Tick(this, EventArgs.Empty); //왜 이렇게 변수를 잡은건지 모름. 물어보기
            getLineDGV1();
            totalSum();
        }
    }
}
```


참조 1개

```
private void MainForm_Load(object sender, EventArgs e)
```

```
{  
    // TODO: 이 코드는 데이터를 'marketDataSet.sales' 테이블에 로드합니다. 필요 시 이 코드를 이동하거나 제거할 수 있습니다.  
    this.salesTableAdapter.Fill(this.marketDataSet.sales);  
}
```

//노선별 검색 리스트박스에 DB 입력

참조 1개

```
public void inputListBox1()
```

```
{  
    DataSet dsRouteNm = selectData("product", "SELECT A.routeNm FROM market.dbo.product A GROUP BY A.routeNm");  
    foreach (DataRow r in dsRouteNm.Tables[0].Rows)  
    {  
        string rtnm = (string)r["routeNm"];  
        listBox1.Items.Add(rtnm);  
    }  
}
```

//휴게소별 검색 리스트박스에 DB 입력

참조 1개

```
public void inputListBox2()
```

```
{  
    DataSet dsstdRestNm = selectData("product", "SELECT A.stdRestNm FROM market.dbo.product A GROUP BY A.stdRestNm");  
    foreach (DataRow s in dsstdRestNm.Tables[0].Rows)  
    {  
        string stnm = (string)s["stdRestNm"];  
        listBox2.Items.Add(stnm);  
    }  
}
```

```
//MSSQL에 접속
참조 3개
public void gconnectDB()
{
    string connect = string.Format("Data Source={0};Initial Catalog = {1};" + //jdbc 데이터형식 (커넥트형식)
        "Persist Security Info = True; User ID=NM; Password=pwer1234"; "localhost", "market");

    conn = new SqlConnection(connect); //SQL 커넥션 생성

    try
    {
        conn.Open(); //db 접속 시도
        label7.Text = "현재 DB 접속 중";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.StackTrace);
    }
}

//특정 데이터베이스에 쿼리문을 넣어서 데이터셋으로 값을 반환
참조 6개
public DataSet gselectData(string tableName, string query)
{
    SqlCommand cmd = (new SqlCommand()); //데이터 베이스 쿼리 명령 생성 (인스턴스? 클래스?)
    cmd.Connection = conn; //접속된 데이터 베이스 연결 전달
    cmd.CommandText = query; //데이터 전체 조회 쿼리 작성
    SqlDataAdapter sda = new SqlDataAdapter(cmd); //쿼리로 나온 값을 읽어와서 담는다?
    DataSet ds = new DataSet(); //조회한 데이터를 담은 객체 생성
    int rowCount = sda.Fill(ds, tableName); //하위 다시 읽어보기 <참조, 기능(입력)>
    return ds; //데이터가 들어간 데이터셋을 반환 (복역오려고!)
}

//MSSQL salesDB에서 구매목록 그리드뷰로 데이터 출력
참조 4개
private void getLineGV10 //해당하는 데이터를 모두 불러오되 한 줄씩 출력
{
    DataSet ds4 = selectData("sales", string.Format("SELECT * FROM market.dbo.sales"));
    //데이터셋 속에 데이터베이스에 쿼리문 넣어서 실행.(구매 왜 여간 string.Format?
    dataGridView5.DataSource = null;
    dataGridView5.DataSource = ds4.Tables[0]; //데이터셋에서 [0]에 해당하는 열로 된 테이블만 제시
}
```

```
//MSSQL salesDB 전체테이블 삭제
참조 1개
private void truncateDB()
{
    selectData("sales", string.Format("truncate table sales"));
    dataGridView5.DataSource = null;
}
```

```
//listBox1
참조 1개
private void listBox1_SelectedIndexChanged(object sender, EventArgs e) //리스트박스에 DB를 넣음
{
    string curItem1 = listBox1.SelectedItem.ToString(); //리스트박스의 현재 선택된 아이템을 curItem에 넣음
    getLineData1(curItem1); //getLineData1 실행. 해당 데이터베이스에서 특정 쿼리를 받아서 실행하여 결과를 반환
    Console.WriteLine(curItem1); //그 결과를 한 줄씩 콘솔에 띄운다.
}

참조 1개
private void getLineData1(string name1) //해당하는 데이터를 모두 불러오되 한 줄씩 출력
{
    DataSet ds1 = selectData("product", string.Format("SELECT * FROM market.dbo.product A WHERE A.routeNm = '{0}'", name1));
    //데이터셋 ds에 데이터베이스에 쿼리문 넣어서 실행.(근데 왜 여긴 string.Format?
    dataGridView1.DataSource = null;
    dataGridView1.DataSource = ds1.Tables[0]; //데이터셋에서 [0]에 해당하는 열로 된 테이블만 제시
}

//listBox2
참조 1개
private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    string curItem2 = listBox2.SelectedItem.ToString(); //리스트박스의 현재 선택된 아이템을 curItem에 넣음
    getLineData2(curItem2); //getLineData2 실행. 해당 데이터베이스에서 특정 쿼리를 받아서 실행하여 결과를 반환
    Console.WriteLine(curItem2); //그 결과를 한 줄씩 콘솔에 띄운다.
}

참조 1개
private void getLineData2(string name2) //해당하는 데이터를 모두 불러오되 한 줄씩 출력
{
    DataSet ds2 = selectData("product", string.Format("SELECT * FROM market.dbo.product A WHERE A.stdRestNm = '{0}'", name2));
    //데이터셋 ds에 데이터베이스에 쿼리문 넣어서 실행.(근데 왜 여긴 string.Format?
    dataGridView2.DataSource = null;
    dataGridView2.DataSource = ds2.Tables[0]; //데이터셋에서 [0]에 해당하는 열로 된 테이블만 제시
}
```

```
참조 1개
private void totalSum()
{
    SqlCommand cmd = new SqlCommand(); //데이터 베이스 쿼리 명령 생성 (인스턴스? 클래스?)
    cmd.Connection = conn; //접속된 데이터 베이스 연결 전달
    cmd.CommandText = "select SUM(CONVERT(INTEGER, foodCost))from market.dbo.sales"; //데이터 전체 조회 쿼리 작성
    Int32 count = (Int32)cmd.ExecuteScalar(); //구매기록 없을 경우에 예외처리 하기
    textBoxSum.Text = count.ToString();
}
```

```

//노선별 검색 top뷰에서 더블클릭하여 under뷰로 특정 열 정보만 이동
참조 1개
private void double_click1(object sender, DataGridViewCellEventArgs e)
{
    int rowIdx = e.RowIndex; //그리드뷰의 선택한 셀 정보를 불러와서 rowIdx에 넣음
    ClickedItemAdd(rowIdx, dataGridView1, dataGridView7, infoList, textBox2);
}

//휴게소별 검색 top뷰에서 더블클릭하여 under뷰로 특정 열 정보만 이동
참조 1개
private void double_click2(object sender, DataGridViewCellEventArgs e)
{
    int rowIdx = e.RowIndex; //리스트 인덱스 가져오기
    ClickedItemAdd(rowIdx, dataGridView2, dataGridView6, infoList2, textBox3);
}

//top뷰에서 더블클릭 했을 때 임시리스트에 row값 입력 후 under뷰에 제시
참조 2개
private void ClickedItemAdd(int SelectedIndex, DataGridView sourceView, DataGridView addView, List<Info> list, TextBox totalCost)
{
    //더블클릭시 이벤트 발생
    //clickcheck++;
    //Console.WriteLine(string.Format("double click {0}", clickcheck));

    DataTable collection = (DataTable)sourceView.DataSource; //데이터 테이블 인스턴스를 가져옴
    DataRow dr = collection.Rows[SelectedIndex]; //데이터 테이블에서 선택된 인덱스 위치의 DataRow를 가져온다.

    Info info = new Info();
    info.foodNm = (string)dr["foodNm"];
    info.foodCost = (string)dr["foodCost"];
    info.stdRestNm = (string)dr["stdRestNm"];
    info.svarAddr = (string)dr["svarAddr"];
    info.routeNm = (string)dr["routeNm"];

    list.Add(info);
    addView.DataSource = null;
    addView.DataSource = list;

    totalPrice(totalCost, list);
}

```

```

//노션별 top뷰 셀을 클릭해서 해당하는 주소 받은 후 gps 불러오기
//text를 입력받아 listBox에 제시하는 방식이 아니므로 Locale 클래스가 필요없었다.
//차후에 다 풀어고쳐서 Locale 없는 버전으로 한 번 만들어 볼 것?
참조 1개
private void CellClick1(object sender, DataGridViewCellEventArgs e)
{
    //if (!webBrowser2.Url.Equals("about: GPS information is missing."))
    //{
    //    webBrowser2.Refresh(); //URL을 새로 기입하라?
    //}
    textBox1.Text = "";
    DataGridViewRow row = dataGridView1.SelectedRows[0]; //선택된 Row 값 가져옴
    string data = row.Cells[3].Value.ToString(); // row의 컬럼(Cells[0]) = name
    //MessageBox.Show(data);

    List<Locale> gpsList = KakaoAPI.Search(data);

    foreach(Locale item in gpsList) //좌표값을 가져온다.
    {
        Console.WriteLine(item.Lng);
        Console.WriteLine(item.Lat);
        Console.WriteLine(item.name);
    }

    try
    {
        object[] pos = new object[] { gpsList[0].Lng, gpsList[0].Lat };
        HtmlDocument hdoc = webBrowser2.Document; //깃허브
        //webBrowser안에 setCenter를 호출한 것.
        //pos가 매개변수가 된 것.
        hdoc.InvokeScript("setCenter", pos); //
    }
    catch (Exception)
    {
        textBox1.Text = "죄송합니다. " + data + "의 GPS 정보가 존재하지 않습니다.";
        //webBrowser2.Navigate("about: GPS information is missing."); //다시 불러오지않는 오류 발생 해결할 것.
        //구매목록 달고나서 구매목록 갔다가 다시 검색페이지 가서 더블클릭하면 달았던 목록 살아나는 버그 해결할 것.
    }
}

```



```
//홈계소셜 top뷰 셀을 클릭해서 해당하는 주소 받은 후 gps 불러오기
참조 1개
private void CellClick2(object sender, DataGridViewCellEventArgs e)
{
    textBox4.Text = "";
    DataGridViewRow row = dataGridView2.SelectedRows[0]; //선택된 Row 값 가져옴
    string data = row.Cells[3].Value.ToString(); // row의 컬럼(Cells[0]) = name

    List<Locale> gpsList = KakaoAPI.Search(data);

    foreach (Locale item in gpsList) //좌표값을 가져온다.
    {
        Console.WriteLine(item.Lng);
        Console.WriteLine(item.Lat);
        Console.WriteLine(item.name);
    }

    try
    {
        object[] pos = new object[] { gpsList[0].Lng, gpsList[0].Lat };
        HtmlDocument hdoc = webBrowser1.Document; //깃허브 index 연결
        ////webBrowser안에 setCenter를 호출한 것.
        ////pos가 매개변수가 된 것.
        hdoc.InvokeScript("setCenter", pos); //index setCenter 실행
    }
    catch (Exception)
    {
        textBox4.Text = "죄송합니다. " + data + "의 GPS 정보가 존재하지 않습니다.";
        //webBrowser1.Navigate("about: GPS information is missing.");
    }
}
```

```
//판매목록에서 그리드뷰 셀 클릭하여 GPS 정보 불러오기
참조 1개
private void CellClick3(object sender, DataGridViewCellEventArgs e)
{
    textBox5.Text = "";
    DataGridViewRow row = dataGridView5.SelectedRows[0]; //선택된 Row 값 가져옴
    string data = row.Cells[3].Value.ToString(); // row의 컬럼(Cells[0]) = name

    List<Locale> gpsList = KakaoAPI.Search(data);

    foreach (Locale item in gpsList) //좌표값을 가져온다.
    {
        Console.WriteLine(item.Lng);
        Console.WriteLine(item.Lat);
        Console.WriteLine(item.name);
    }

    try
    {
        object[] pos = new object[] { gpsList[0].Lng, gpsList[0].Lat };
        HtmlDocument hdoc = webBrowser3.Document; //깃허브 index 연결
        ////webBrowser안에 setCenter를 호출한 것.
        ////pos가 매개변수가 된 것.
        hdoc.InvokeScript("setCenter", pos); //index setCenter 실행
    }
    catch (Exception)
    {
        textBox5.Text = "죄송합니다. " + data + "의 GPS 정보가 존재하지 않습니다.";
        //webBrowser1.Navigate("about: GPS information is missing.");
    }
}
```

```
//노션에 under류에 담은 리스트에서 항목 삭제하는 더블클릭
참조 1개
private void gubleClickCancel1(object sender, DataGridViewCellEventArgs e)
{
    int rowIdx = e.RowIndex; //리스트 인덱스 가져오기
    ClickCancel(rowIdx, textBox2, dataGridView7, infoList);
}

//음게소일 under류에 담은 리스트에서 항목 삭제하는 더블클릭
참조 1개
private void gubleClickCancel2(object sender, DataGridViewCellEventArgs e)
{
    int rowIdx = e.RowIndex; //리스트 인덱스 가져오기
    ClickCancel(rowIdx, textBox3, dataGridView6, infoList2);
}

//임시 리스트 내의 비용을 삭제하는 함수
참조 2개
private void ClickCancel(int SelectedIndex, TextBox textbox, DataGridView dataGridView, List<Info> list)
{
    list.RemoveAt(SelectedIndex); //지움 리스트 가져와서 해당 row인덱스 삭제
    dataGridView.DataSource = null; //새로고침
    dataGridView.DataSource = list; //리스트 그리드뷰에 재지

    totalPrice(textbox, list); //합계 계산 refresh
}

//담은 목록 전체 삭제하는 버튼
참조 2개
private void gllCancel(TextBox textbox, DataGridView dataGridView, List<Info> list)
{
    list.Clear();
    dataGridView.DataSource = null;
    dataGridView.DataSource = list;

    totalPrice(textbox, list);
}
```

```
//담은 목록 금액 합계함수
참조 3개
private void totalPrice(TextBox textBox, List<Info> list)
{
    int sum = 0;

    foreach (Info item in list)
    {
        sum += Convert.ToInt32(item.foodCost);
    }
    textBox.Text = sum.ToString();
}

//구매목록 금액 합계 텍스트박스
참조 1개
private void textBoxSum_TextChanged(object sender, EventArgs e)
{
    //totalPrice(textBox5, )
}
```

```
//판매내역 DB로 구매내역 입력
참조 2개
public static void InputSales(string foodNm, string foodCost, string stdRestNm, string svarAddr,
    string routeNm)
{
    SqlConnection conn = null;
    try
    {
        string connect = string.Format("Data Source={0};Initial Catalog = {1};" +
            "Persist Security Info = True; User ID=NM; Password=naver1234", "localhost", "market");
        conn = new SqlConnection(connect); //SQL 커넥션 생성
        conn.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn; //어디에 커맨드 보낼지 지정
        string cmdstr = string.Format("IF EXISTS(SELECT foodNm FROM sales WHERE foodNm='{0}') " +
            "BEGIN UPDATE sales SET foodCost = '{1}', stdRestNm='{2}', svarAddr='{3}', routeNm='{4}' " +
            "WHERE foodNm='{0}' END ELSE BEGIN INSERT INTO " +
            "sales (foodNm,foodCost,stdRestNm, svarAddr, routeNm) values ('{0}','{1}', '{2}', '{3}', " +
            "'{4}') END",
            foodNm, foodCost, stdRestNm, svarAddr, routeNm);
        //MessageBox.Show(cmdstr);
        cmd.CommandText = cmdstr;
        cmd.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        int i = 0;
        //System.Windows.Forms.MessageBox.Show(e.Message + "insert");
        //System.Windows.Forms.MessageBox.Show(e.StackTrace + "insert");
        //DataManager.printLog("select" + e.StackTrace);
        return;
    }
    finally
    {
        conn.Close(); //db 연결 해제
    }
}
```

```
//노선별 검색 계산하기 버튼
참조 1개
private void button4_Click(object sender, EventArgs e)
{
    connectDB();
    foreach (Info item in infoList)
    {
        InputSales(item.foodNm, item.foodCost, item.stdRestNm, item.svarAddr,
            item.routeNm);
    }
    dataGridView7.DataSource = null;
    getLineDGV1();
}

//휴게소별 검색 계산하기 버튼
참조 1개
private void button7_Click(object sender, EventArgs e)
{
    connectDB();
    foreach (Info item in infoList2)
    {
        InputSales(item.foodNm, item.foodCost, item.stdRestNm, item.svarAddr,
            item.routeNm);
    }
    dataGridView6.DataSource = null;
    getLineDGV1();
}

//구매목록 전체삭제 버튼
참조 1개
private void button1_Click(object sender, EventArgs e)
{
    truncateDB();
    getLineDGV1();
    MessageBox.Show("당긴 물품이 없습니다.");
}
```



```
//노선별 검색 계산하기 버튼
참조 1개
private void button4_Click(object sender, EventArgs e)
{
    connectDB();
    foreach (Info item in infoList)
    {
        inputSales(item.foodNm, item.foodCost, item.stdRestNm, item.svarAddr,
            item.routeNm);
    }
    dataGridView7.DataSource = null;
    getLineDGV1();
}

//휴게소별 검색 계산하기 버튼
참조 1개
private void button7_Click(object sender, EventArgs e)
{
    connectDB();
    foreach (Info item in infoList2)
    {
        inputSales(item.foodNm, item.foodCost, item.stdRestNm, item.svarAddr,
            item.routeNm);
    }
    dataGridView6.DataSource = null;
    getLineDGV1();
}

//구매목록 전체삭제 버튼
참조 1개
private void button1_Click(object sender, EventArgs e)
{
    truncateDB();
    getLineDGV1();
    MessageBox.Show("담긴 물품이 없습니다.");
}
```

```
//구매목록 전체삭제 버튼
참조 1개
private void button1_Click(object sender, EventArgs e)
{
    truncateDB();
    getLineDGV1();
    MessageBox.Show("담긴 물품이 없습니다.");
}

//노선별 검색 전체취소 버튼
참조 1개
private void button3_Click(object sender, EventArgs e)
{
    allCancel(textBox2, dataGridView7, infoList);
    MessageBox.Show("선택메뉴를 모두 비웠습니다.");
}

//휴게소별 검색 전체취소 버튼
참조 1개
private void button8_Click(object sender, EventArgs e)
{
    allCancel(textBox3, dataGridView6, infoList2);
    MessageBox.Show("선택메뉴를 모두 비웠습니다.");
}

//현재 시간 표시
참조 2개
private void timer1_Tick(object sender, EventArgs e)
{
    label2.Text = DateTime.Now.ToString($"{yyyy}년 MM월 dd일 HH시 mm분 ss초");
}

//창 끄기
참조 1개
private void button2_Click(object sender, EventArgs e)
{
    Close();
}
```

```

참조 3개
public class KakaoAPI
{
    //클래스 메소드 만들
    //KakaoAPI.Search("대구");
    참조 3개
    public static List<Locale> Search(string text)
    {
        List<Locale> List = new List<Locale>();

        string site = "https://dapi.kakao.com/v2/local/search/keyword.json";
        string query = $"{site}?query={text}";
        string restApiKey = "85104fd0d7225ae569320887a41509b";
        string Header = $"KakaoAK {restApiKey}";

        //Using System.Net
        WebRequest request = WebRequest.Create(query);
        request.Headers.Add("Authorization", Header);

        //응답받기
        WebResponse response = request.GetResponse();
        //Using System.IO
        Stream stream = response.GetResponseStream();
        StreamReader reader = new StreamReader(stream, Encoding.UTF8);
        string json = reader.ReadToEnd();

        JavaScriptSerializer js = new JavaScriptSerializer();
        //누겟에서 뉴턴 설치 후 alt+enter 해서 Using 추가
        //로컬정보도 json에서 가져오기 때문에 ??

        //dynamic = js에서의 let같은 것
        dynamic doc = js.Deserialize<dynamic>(json);
        dynamic docs = doc["documents"];
        object[] buf = docs;
        int length = buf.Length;
        for (int i = 0; i < length; i++)
        {
            string lname = docs[i]["place_name"];
            double x = double.Parse(docs[i]["x"]);
            double y = double.Parse(docs[i]["y"]);
            List.Add(new Locale(lname, x, y));
        }

        return List;
    }
}

```

```
namespace SnacksOnHighway
{
    참조 13개
    public class Info
    {
        참조 3개
        public string foodNm { get; set; } //음식 이름
        참조 4개
        public string foodCost { get; set; } //음식 가격
        참조 3개
        public string stdRestNm { get; set; } //휴게소 이름
        참조 3개
        public string syarAddr { get; set; } //휴게소 주소
        참조 3개
        public string routeNm { get; set; } //노선명
        참조 0개
        public string totalPrice { get; set; } //가격 합계
        참조 0개
        public DateTime sellTime { get; set; } //구매시각
    }
}
```

```

namespace SnacksOnHighway
{
    참조 4개
    public partial class IntroForm : Form
    {
        //static string requestURL = "https://data.ex.co.kr/openapi/restinfo/restBestfood_list?key=9858367230&type=json&numOfRows=99&pageNo=1.";

        private SqlConnection conn = null; //db 접속변수
        private SqlDataAdapter sda = null;
        private DataSet ds = null;
        private DataTable dt = null;
        //getset은 일반변수에서만 쓰면 된다.
        //클래스 변수에서는 사용 안 함?
        //int float double byte 이렇게 일반변수
        //SqlConnection conn 에서 conn은 사실 변수가 아니다.
        //변수는 값이 계속 변경시켜 저장하는 거라 변수인데
        //conn은 한 번 선언해서 계속 변경없이 사용한다.
        //이것을 참조 또는 인스턴스라고 한다.

        //일단 클래스로 따로 빼는 건 시도하지 말자.. 어렵다.

        //DataTable table = new DataTable(); //table 이라는 인스턴스 메소드 생성

        참조 1개
        public IntroForm()
        {
            InitializeComponent();
            connectDB();
            //json_data();
        }

        참조 1개
        private void button1_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}

```

```

참조 1개
private void button6_Click(object sender, EventArgs e)
{
    mainForm lockusForm = new mainForm();
    lockusForm.ShowDialog();
    this.Close();
}

//MSQL에 접속
참조 1개
public void connectDB()
{
    string connect = string.Format(
        ("Data Source={0};Initial Catalog = {1};" + //json 데이터형식 (키값유형)
        "Persist Security Info = True;" +
        "User ID=HR; Password=wer1234;" + "localhost"; "werket");

    conn = new SqlConnection(connect); //접속 정보를 팔아서 MSQL 커넥션 생성 (conn을 위해서 선언)

    try
    {
        conn.Open(); // db 접속 시도
        label2.Text = "현재 DB 접속 중"; //접속상태 표시
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.StackTrace);
    }
}

참조 1개
private string CreateRequestURLFromPage(int page) //특정 페이지를 불러오는 가정으로 변수명을 page로 설정.
{
    string requestURL =
        "https://data-ex.co.kr/openapi/rest/info/restBestFood_list?category=2&page=" + Convert.ToString(page);
    //공공데이터 4차: 마지막 숫자가 페이지 넘버라서 변수로 할당(문자열로 받기)
    return requestURL;
    //값을 반환하여 넘긴다.
}

```

참조 0개

```

public void json_data()
{
    for(int i = 0; i < 44; i++)
    {
        string reqURL = CreateRequestURLFromPage(i + 1);
        requestProc(reqURL);
    }
}

```

```

//공통 코드
public void requestProc(string url)
{
    Console.WriteLine("Page Uri => " + url);

    WebRequest request = WebRequest.Create(url); //데이터를 요청할 주소값 입력
    request.Method = "GET"; //HTTP GET 방식으로 호출
    request.ContentType = "application/json"; //요청 컨텐트 형식은 json

    using (WebResponse response = request.GetResponse()) //HTTP Response를 값을 객체 선언
    using (Stream stream = response.GetResponseStream()) //Response로부터 Stream 객체 생성
    using (StreamReader reader = new StreamReader(stream)) //StreamReader로 데이터 읽어오기 위한 객체 생성
    {
        string data = reader.ReadToEnd(); //StreamReader로 Stream 데이터를 끝까지 받아서 String으로 반환받음
        var obj = JObject.Parse(data); //올라온 데이터가 Json Object로 시작하기 때문에 JObject로 Parse
        var list = (JArray)obj["list"]; //데이터 목록값이 "list" key를 가지고 같은 Json List로 되어있으므로
        //JArray로 변환하여 list 변수에 담음

        Console.WriteLine("=====list=====");
        foreach (JObject item in list) //목록을 반복하면서 데이터 값이 담긴 JObject을 하나씩 처리함
        {
            string foodNm = (string)item["foodNm"];
            string foodCost = (string)item["foodCost"];
            string stdRestNm = (string)item["stdRestNm"];
            string svarAddr = (string)item["svarAddr"];
            string routeNm = (string)item["routeNm"];
            string etc = (string)item["etc"];
            string foodMaterial = (string)item["foodMaterial"];
            string recommendyn = (string)item["recommendyn"];
            string seasonMenu = (string)item["seasonMenu"];
            string bestfoodyn = (string)item["bestfoodyn"];
            string premiumyn = (string)item["premiumyn"];
            string lsttAltrOtta = (string)item["lsttAltrOtta"];

            //Console.WriteLine(string.Format("FoodNm => {0} | foodCost => {1} | svarAddr => {2} | " +
            //    " | routeNm => {3} | etc => {4} | foodMaterial => {5} | recommendyn => {6} | " +
            //    " | seasonMenu => {7} | bestfoodyn => {8} | premiumyn => {9} | lsttAltrOtta => {10} " +
            //    " | foodNm, foodCost, svarAddr, routeNm, etc, foodMaterial, recommendyn,
            //    seasonMenu, bestfoodyn, premiumyn, lsttAltrOtta); //값 가져오는지 콘솔에 출력

            inputProduct(foodNm, foodCost, stdRestNm, svarAddr,
                routeNm, etc, foodMaterial,
                recommendyn, seasonMenu, bestfoodyn,
                premiumyn, lsttAltrOtta);
        }

        DataTable dt = JsonConvert.DeserializeObject<DataTable>(list.ToString());
        //Json List를 JsonString으로 출력하여 전달 -> DataTable 타입으로 Deserialize 수법 -> DataTable에 데이터가 담김
        //dataGridView1.DataSource = dt; //데이터 바인딩
    }
}

```



```

//MySQL에 데이터 입력하기
참조 1개
public static void InputProduct(string foodNm, string foodCost, string stdRestNm, string sverAddr, string routeNm,
string etc, string foodMaterial, string recommendyn, string seasonMenu, string bestFoodyn,
string premiumyn, string lstttwAltrDtta)
{
    SqlConnection conn = null;
    try
    {
        if(etc != null)
        {
            etc = etc.Replace("'", "");
        }

        if(foodMaterial != null)
        {
            foodMaterial = foodMaterial.Replace("'", "");
        }

        string connect = string.Format("Data Source={0};Initial Catalog = {1};" +
            "Persist Security Info = True; User ID=NM; Password=naver1234; ", "localhost", "market");
        conn = new SqlConnection(connect); //SQL커넥션 생성
        conn.Open();

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn; //어디에 커멘드 보낼지 지정
        string cmdstr = string.Format("IF EXISTS(SELECT foodNm FROM product WHERE foodNm='{0}')" +
            "BEGIN UPDATE product SET foodCost = '{1}', stdRestNm='{2}', sverAddr='{3}', routeNm='{4}'," +
            "etc='{5}', foodMaterial='{6}', recommendyn='{7}', seasonMenu='{8}', bestFoodyn='{9}'," +
            "premiumyn='{10}', lstttwAltrDtta='{11}' WHERE foodNm = '{0}' END ELSE BEGIN INSERT INTO " +
            "product (foodNm, foodCost, stdRestNm, sverAddr, routeNm, etc, foodMaterial, recommendyn," +
            "seasonMenu, bestFoodyn, premiumyn, lstttwAltrDtta) values ('{0}', '{1}', '{2}', '{3}', " +
            "'{4}', '{5}', '{6}', '{7}', '{8}', '{9}', '{10}', '{11}') END";
            foodNm, foodCost, stdRestNm, sverAddr, routeNm, etc, foodMaterial, recommendyn,
            seasonMenu, bestFoodyn, premiumyn, lstttwAltrDtta);
        //MessageBox.Show(cmdstr);
        cmd.CommandText = cmdstr;
        cmd.ExecuteNonQuery();

    }
    catch (Exception e)
    {
        int j = 0;
        //System.Windows.Forms.MessageBox.Show(e.Message + "Insert");
        //System.Windows.Forms.MessageBox.Show(e.StackTrace + "Insert");
        //DataManager.printLog("select" + e.StackTrace);
        return;
    }
    finally
    {
        conn.Close(); //db 연결 해제
    }
}

```

DESKTOP-C3JUME2 (SQL Server 15.0.2000.5 - N/A)

데이터베이스

- 시스템 데이터베이스
- 데이터베이스 스냅샷
- market
 - 데이터베이스 다이어그램
 - 테이블
 - 시스템 테이블
 - FileTables
 - 외부 테이블
 - 그래프 테이블
 - dbo.product
 - 문
 - 기
 - 제약 조건
 - 트리거
 - 인덱스
 - 통계
- dbo.sales
 - 업
 - 키
 - PK_sales
 - 제약 조건
 - 트리거
 - 인덱스
 - 통계
- dbo.storage
 - 뷰
 - 외부 리소스
 - 동역어
 - 프로그래밍 기능
 - Service Broker
 - 스토리지
 - 보안

***** SSMS의 SelectTopNRows 명령 스크립트 *****/

```

SELECT TOP (1000) [foodNm]
, [foodCost]
, [stdRestNm]
, [svarAddr]
, [routeNm]
, [etc]
, [foodMaterial]
, [recommendyn]
, [seasonMenu]
, [bestfoodyn]
, [premiumyn]
, [lsttmAltrDttm]
FROM [market].[dbo].[product]
  
```

133 %

결과

	foodNm	foodCost	stdRestNm	svarAddr	routeNm	etc
1	(농부)농부부대찌개	8500	남성주함외(광원)육계소	경북 상주군 선남면영포리 880-5	중부내륙선	농부부대찌개 브랜드의 대표 음식
2	(농부)왕갈비갈판	8500	남성주함외(광원)육계소	경북 상주군 선남면영포리 880-5	중부내륙선	농부부대찌개 브랜드로 왕갈비와 불맛을 살린 갈판
3	(농부)새우볶음밥	8000	남성주함외(광원)육계소	경북 상주군 선남면영포리 880-5	중부내륙선	농부부대찌개 브랜드 중 새우로 만든 볶음밥
4	(농부)왕갈비치볶음밥	8000	남성주함외(광원)육계소	경북 상주군 선남면영포리 880-5	중부내륙선	농부부대찌개 브랜드 중 갈비와 갈비의 조합의 볶음밥
5	(농부)농부부대찌개	8500	남성주함외(광원)육계소	경북 상주군 선남면영포리 880-5	중부내륙선	농부부대찌개 브랜드로 치즈와 부대찌개의 조합으로 만든 찌개
6	(매콤 비빔국) 떡도그	2000	내원천육계소	강원 인제군 삼남면 하남리 372번지	서울양양선	
7	(후)신라면	4500	김천(부산)육계소	전북 김천시 봉소면신촌리 730-2	경부선	매콤 후, 한국인이 사랑하는 신라면을 끓여드립니다.
8	(실속)남원시래기국밥	5500	지리산(대구)육계소	전북 남원시 어연면아곡리 산83	광주대곡선	남도 시래기의 구수한맛에 얹혀있는 남원시래기국밥!! 고소한 들깨가루와
9	(실속)원장찌개	5500	예산(당진)육계소	충남 예산군 신암면서계알리 672-3	당진열덕선	(실속)원장찌개
10	(실속)원장찌개(궁기밥)	6500	전통산(세원)육계소	충북 충주시 신척면팔곡리세원교로도로 105	평덕세원선	
11	(실속)바섯음장찌개	5500	보성복자(영암)육계소	전남 보성군 미력면화발리 산 195-7	남해선	
12	(실속)우동	3000	남성주함외(광원)육계소	경북 상주군 선남면영포리 880-5	중부내륙선	부담없는 가격과 실속 우동
13	(실속)왕바다바지락탕	6500	문막(인천)육계소	강원 원주시 문막읍보전리	영동선	죽배기 듬뿍 바지락가득 담아 만들어진 시원한맛이 일품인 바지락탕

개체 탐색기

DESKTOP-C3JUME2 (SQL Server 15.0.2000.5 - NM ^

- 데이터베이스
 - 시스템 데이터베이스
 - 데이터베이스 스냅샷
 - market
 - 데이터베이스 다이어그램
 - 테이블
 - 시스템 테이블
 - FileTables
 - 외부 테이블
 - 그래프 테이블
 - dbo.product
 - dbo.sales
 - 열
 - 키
 - PK_sales
 - 제약 조건
 - 트리거
 - 인덱스
 - 통계
 - dbo.storage
 - 뷰
 - 외부 리소스
 - 동의를
 - 프로그래밍 기능
 - Service Broker
 - 스토리지
 - 보안
 - 보안
 - 서버 개체

SQLQuery75.sql - ...2.market (NM (83))

SQLQuery74.sql - ...2.market (NM (84))

SQLQuery73.sql - ...2

```

/***** SSMS의 SelectTopNRows 명령 스크립트 *****/
SELECT TOP (1000) [foodNm]
      , [foodCost]
      , [stdRestNm]
      , [svarAddr]
      , [routeNm]
FROM [market].[dbo].[sales]
  
```

133 %

결과 메시지

	foodNm	foodCost	stdRestNm	svarAddr	routeNm
1	국산 우령 우리콩원장찌개	7500	송산휴게소	경기 화성시 송산면고정리 349-26	평택시흥선
2	맛나채육정식	9000	송산휴게소	경기 화성시 송산면고정리 349-26	평택시흥선
3	송산 포도호두과자	11000	송산휴게소	경기 화성시 송산면고정리 349-26	평택시흥선

프로젝트 개발결과



현재 DB 접속 중



Snacks On Highway

고속도로 휴게소 메뉴검색

입장

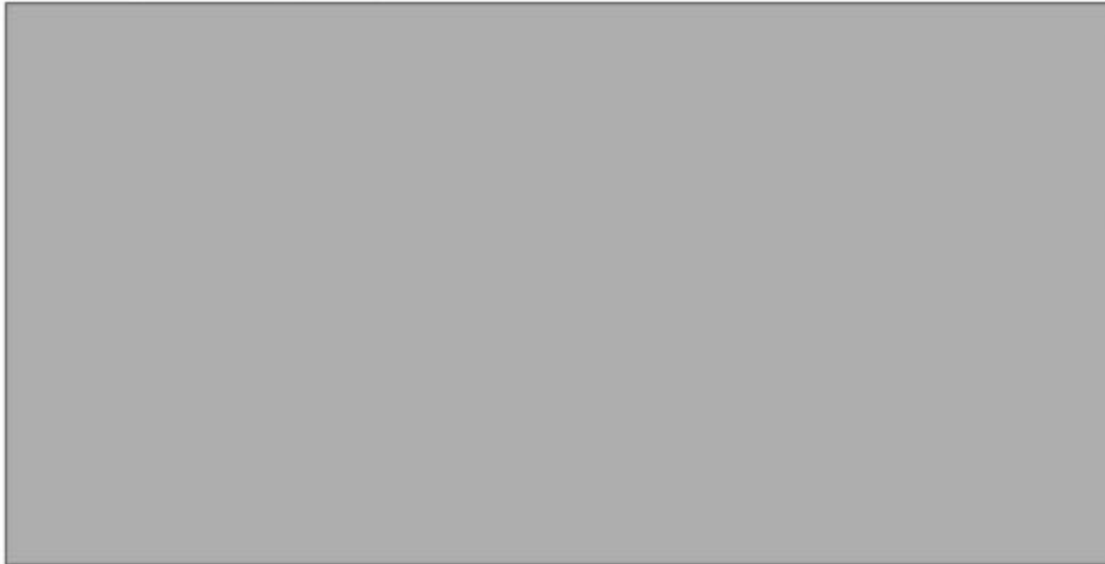
Contact Us.



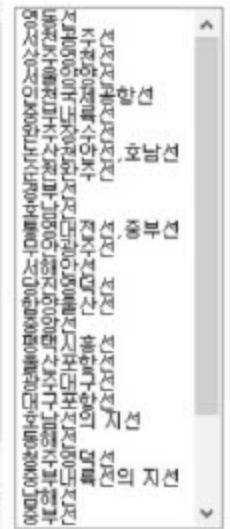
노선별로 검색기능

현재 DB 접속 중

노선별로 검색 휴게소별로 검색 구매내역



노선명



합계

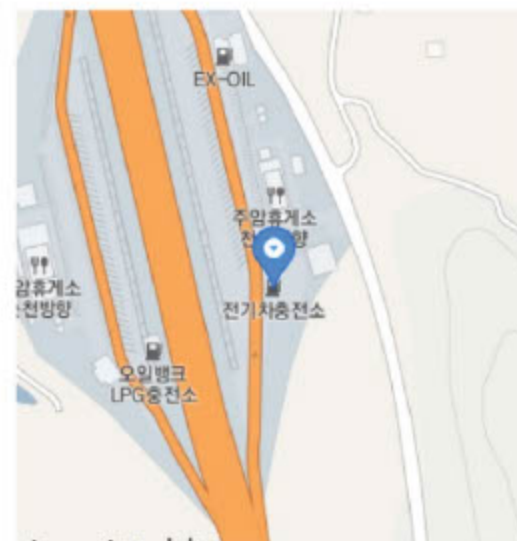
전체취소

담기

2022년 12월 15일 03시 46분 31초

노선별로 검색 휴게소별로 검색 구매내역

foodNm	foodCost	stdRestNm	svarAddr	routeNm	etc	food
(착한)삼채된장...	6500	백양사(순천)휴...	전라남도 장성...	호남선	삼채된장찌개	삼채
(코바코)옛날돈...	8000	주암(순천)휴게소	전남 순천시 주...	호남선		
(코바코)치즈돈...	9000	주암(순천)휴게소	전남 순천시 주...	호남선		
가자미들깨미...	6500	곡성(천안)휴게소	전남 곡성군 곶...	호남선	6500원으로 구...	
갈비탕	8000	순천(순천)휴게소	전남 순천시 서...	호남선		
고등어구이	9000	백양사(순천)휴...	전라남도 장성...	호남선	고온의 화덕에...	고등
고등어구이정식	8500	여산(순천)휴게소	전북 익산시 여...	호남선		
곡성 모듬파스	12000	곡성(순천)휴게소	전남 곡성군 곶...	호남선		
곡성매실액제...	10000	곡성(천안)휴게소	전남 곡성군 곶...	호남선	달콤한 매실액...	
▶ 김치 된장찌개	8500	주암(천안)휴게소	전남 순천시 주...	호남선		
김치볶음밥	7000	여산(순천)휴게소	전북 익산시 여...	호남선		
김치우동	6500	여산(순천)휴게소	전북 익산시 여...	호남선		



노선명

합계

전체취소

담기

노선별로 검색 홍계소별로 검색 구매내역

	foodNm	foodCost	stdRestNm	svarAddr	routeNm	etc	food
	(학한)삼채된장...	6500	백양사(순천)휴...	전라남도 장성...	호남선	삼채된장찌개	삼채
	(코바코)옛날돈...	8000	주암(순천)휴게소	전남 순천시 주...	호남선		
	(코바코)치즈돈...	9000	주암(순천)휴게소	전남 순천시 주...	호남선		
	가자미물때미...	6500	곡성(천안)휴게소	전남 곡성군 곶...	호남선	6500원으로 구...	
	갈비탕	8000	순천(순천)휴게소	전남 순천시 서...	호남선		
	고등어구이	9000	백양사(순천)휴...	전라남도 장성...	호남선	고온의 화덕에...	고등
	고등어구이정식	8500	여산(순천)휴게소	전북 익산시 여...	호남선		
▶	곡성 모듬파스	12000	곡성(순천)휴게소	전남 곡성군 곶...	호남선		
	곡성매실액제...	10000	곡성(천안)휴게소	전남 곡성군 곶...	호남선	달콤한 매실액...	
	김치 덮지찌개	8500	주암(천안)휴게소	전남 순천시 주...	호남선		
	김치볶음밥	7000	여산(순천)휴게소	전북 익산시 여...	호남선		
	김치우동	6500	여산(순천)휴게소	전북 익산시 여...	호남선		

	foodNm	foodCost	stdRestNm	svarAddr	routeNm	totalPrice	sellTm
▶	김치 덮밥찌개	8500	주암(천안)휴게소	전남 순천시 주...	호남선		
	곡성매실액제...	10000	곡성(천안)휴게소	전남 곡성군 곡...	호남선		
	곡성 모듬까스	12000	곡성(순천)휴게소	전남 곡성군 곡...	호남선		



노선별로 검색 휴게소별로 검색 구매내역

[illegible]

합계
12000

전체취소

담기

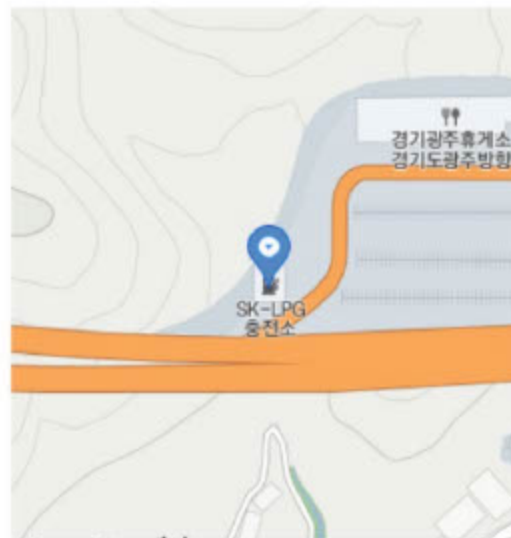
노선별로 검색 휴게소별로 검색 구매내역

	foodNm	foodCost	stdRestNm	svarAddr	routeNm	etc	food
	(척합)삼채된장...	6500	백양사(순천)휴...	전라남도 장성...	호남선	삼채된장찌개	삼채
	(코바코)옛날돈...	8000	주암(순천)휴게소	전남 순천시 주...	호남선		
	(코바코)치즈돈...	9000	주암(순천)휴게소	전남 순천시 주...	호남선		
	가자미물탕미...	6500	곡성(천안)휴게소	전남 곡성군 곡...	호남선	6500원으로 구...	
	갈비탕	8000	순천(순천)휴게소	전남 순천시 서...	호남선		
	고등어구이	9000	백양사(순천)휴...	전라남도 장성...	호남선	고운의 화덕에...	고등
	고등어구이정식	8500	여산(순천)휴게소	전북 익산시 여...	호남선		
▶	곡성 모듬파스	12000	곡성(순천)휴게소	전남 곡성군 곡...	호남선		
	곡성매실액제...	10000	곡성(천안)휴게소	전남 곡성군 곡...	호남선	달콤한 매실액...	
	김치 덮지찌개	8500	주암(천안)휴게소	전남 순천시 주...	호남선		
	김치볶음밥	7000	여산(순천)휴게소	전북 익산시 여...	호남선		
	김치우동	6500	여산(순천)휴게소	전북 익산시 여...	호남선		



노선별로 검색 휴게소별로 검색 구매내역

	foodNm	foodCost	stdRestNm	svarAddr	routeNm	etc	foodM
	콘드레밥과 용...	15000	광주(광주)휴게소	경기 광주시곤...	중앙선		
	소매리곰탕	11000	광주(광주)휴게소	경기 광주시곤...	중앙선		
	양곰탕	10000	광주(광주)휴게소	경기 광주시곤...	중앙선		
	자맛육계장칼...	4500	광주(광주)휴게소	경기 광주시곤...	중앙선		
▶	코다리만두국	7500	광주(광주)휴게소	경기 광주시곤...	중앙선		



노선명

합계

전체회소

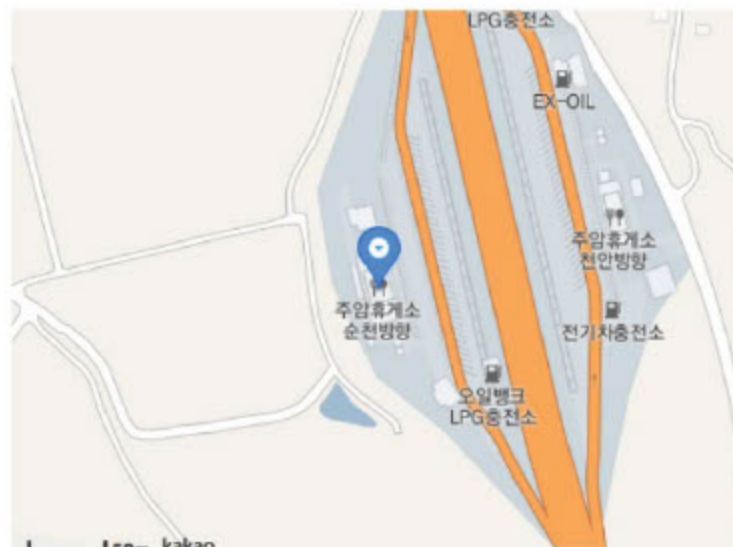
담기

구매내역 조회기능

현재 DB 접속 중

노선별로 검색 휴게소별로 검색 구매내역

	foodNm	foodCost	stdRestNm	svarAddr	routeNm
▶	고등어구이정식	8500	여산(순천)휴게소	전북 익산시 여...	호남선
	곡성 모듬파스	12000	곡성(순천)휴게소	전남 곡성군 곁...	호남선



좌

비우기

결재하기

2022년 12월 15일 03시 53분 39초

구매내역 삭제기능

현재 DB 접속 중

노선별로 검색 휴게소별로 검색 구매내역

foodNm	foodCost	stdRestNm	svarAddr	routeNm
▶ 고등어구이정식	8500	여산(순천)휴게소	전북 익산시 여...	호남선
곡성 모듬파스	12000	곡성(순천)휴게소	전남 곡성군 곁...	호남선



출역

20500

비우기

결재하기

2022년 12월 15일 04시 15분 21초

구매내역 삭제기능

현재 DB 접속 중

노선별로 검색 휴게소별로 검색 구매내역

foodNm	foodCost	stdRestNm	svarAddr	routeNm

×

당긴 물품이 없습니다.

확인



총액

20500

비우기

결제하기

2022년 12월 15일 04시 15분 44초

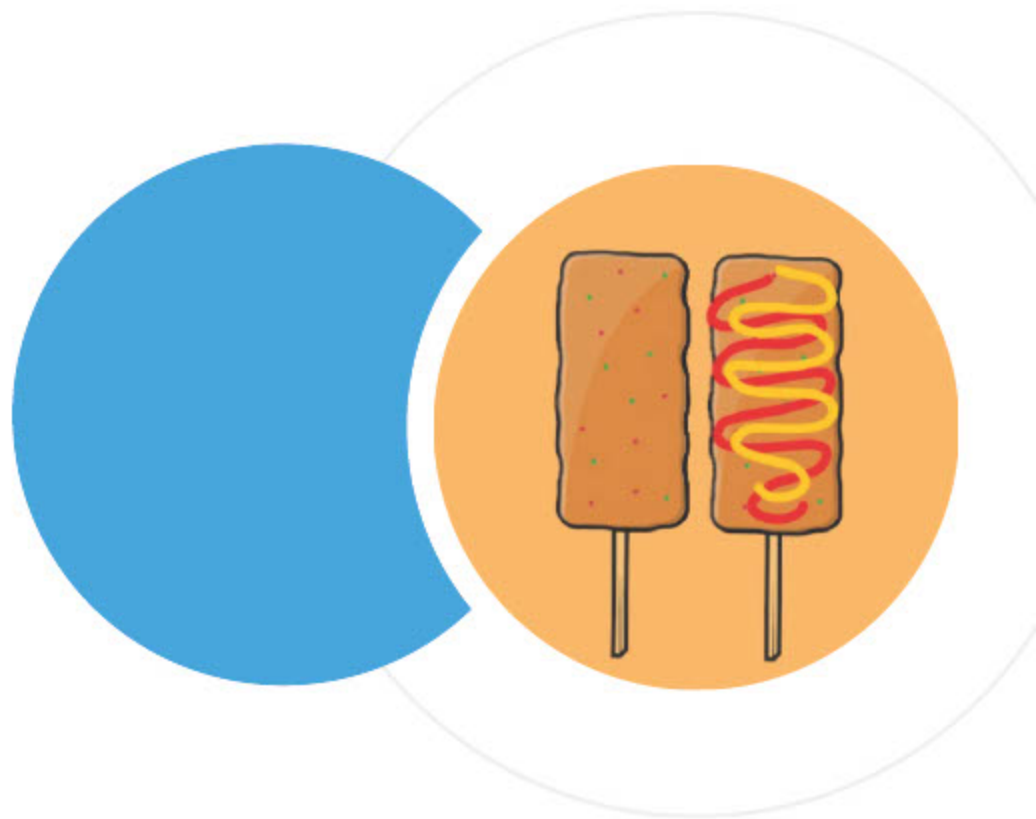
자체평가

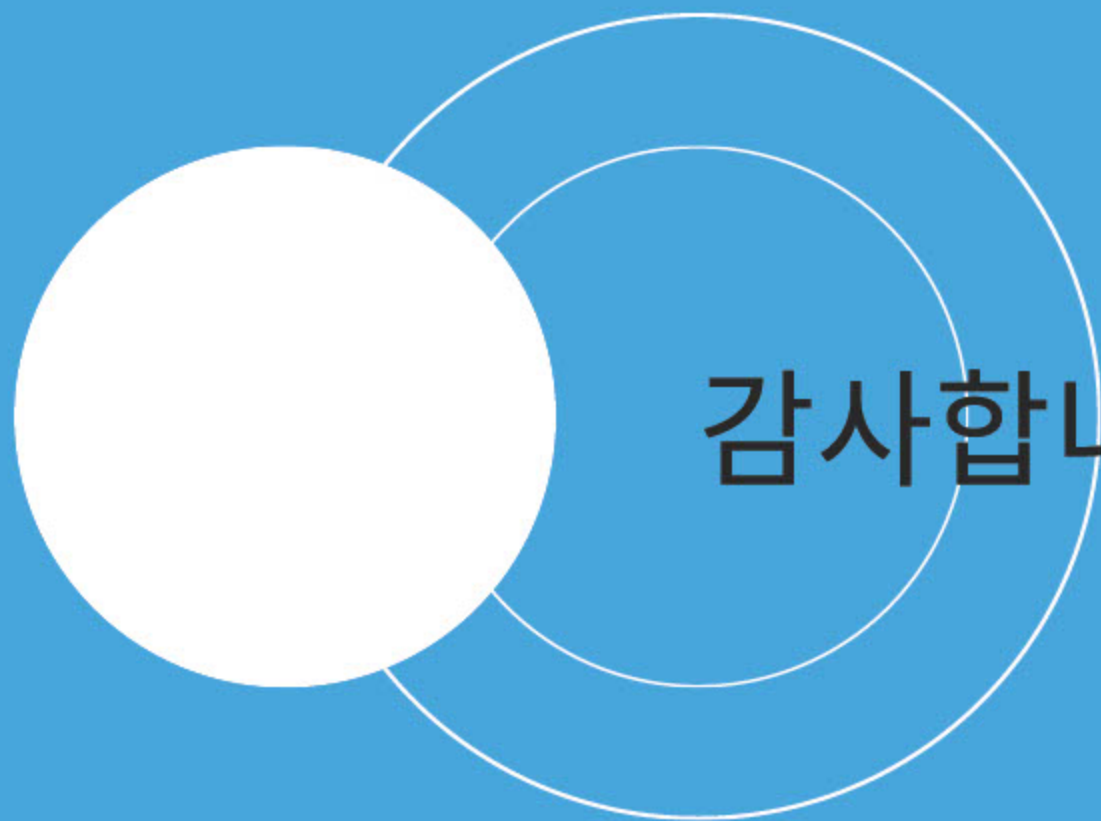
차우 수정해야 할 오류를 개발 중간에 다수 발견

-GPS 정보 불러오는 과정 중에 지도정보 새로고침이 되지 않는 경우 발생 => 차후 URL 다시 불러오는 방식으로 수정 예정

-구매목록 담고나서 구매목록 갔다가 다시 검색으로 돌아가서 물품을 더블클릭하면 담았던 목록 살아나는 오류 => 파악 후 차후 수정 예정

수업시간에 배운 코드에서 변형이 많이 필요한 상황이 많았습니다. 그 만큼 의도만큼 프로그램이 실행되지 않는 경우가 많았고 오류를 파악하기 위해 구글링도 많이 필요했습니다. 그런 노력에도 불구하고 모든 것이 원활하게 이루어진 프로젝트는 아니라서 아쉬움이 남았습니다. 차후에 수정을 통해 좀 더 완벽한 프로젝트로 완성 할 것입니다.





감사합니다!

